

**AMENDMENTS TO THE CLAIMS**

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of claims:

1. (currently amended) A method implemented at least partially in a programmed computer for software application development suitable for processing input files with varying file formats or data structures, the method comprising:

characterizing the file format and data structure of at least one known input file type using the programmed computer; and

creating a library comprising a plurality of jobs ~~configured~~ to perform predetermined functions using the programmed computer, wherein the library of jobs comprises at least one job ~~configured~~ to read the file format and data structure of the known input file and convert that input file to a tab delimited file format or data structure that is common to the plurality of jobs, at least one job ~~configured~~ for extracting data from a file, at least one job ~~configured~~ for archiving files, at least one job ~~configured~~ for loading files, at least one job ~~configured~~ for messaging, at least one job ~~configured~~ for transforming data in a file, at least one job ~~configured~~ for validating data in a file, and further wherein each job includes an indicator of job termination success or job termination failure, and the plurality of jobs can be linked ~~are configured for linking~~ according to the indicator of job termination success or job termination failure.

2-7. (cancelled)

8. (currently amended) A method according to claim 1, further comprising:  
creating a job stream library using the programmed computer, the job stream library comprising at least one job stream, the job stream comprising individual jobs from the library of jobs, and the at least one job stream in the job stream library including an indicator of job stream termination, wherein the indicator is either job stream termination success or job stream termination failure.

9. (currently amended) A method according to claim 1, further comprising adding jobs to the library of jobs using the programmed computer.

10. (currently amended) A system implemented at least partially in a programmed computer for software application development suitable for processing input files with varying file formats or data structures, the system comprising a processor with:

computer program means for characterizing the file format and data structure of at least one known input file type; and

computer program means for creating a library comprising a plurality of jobs ~~configured~~ to perform predetermined functions, wherein the library of jobs comprises at least one job ~~configured~~ to read the file format and data structure of the known input file and convert that input file to a tab delimited file format or data structure that is common to the plurality of jobs, at least one job ~~configured~~ for extracting data from a file, at least one job ~~configured~~ for archiving files, at least one job ~~configured~~ for loading files, at least one job ~~configured~~ for messaging, at least one job ~~configured~~ for transforming data in a file, at least one job ~~configured~~ for validating data in a file, and further wherein each job includes an indicator of job termination success or job

termination failure, and the plurality of jobs can be linked ~~are configured for linking~~ according to the indicator of job termination success or job termination failure.

11. (cancelled)

12. (currently amended) A computer-executable program that is tangibly embodied on a computer-readable medium as having computer executable software code stored ~~thereon~~, the code for software application development suitable for processing input files with varying file formats or data structures, the code comprising:

code to characterize the file format and data structure of at least one known input file type; and

code to create a library comprising a plurality of jobs ~~configured~~ to perform predetermined functions, wherein the library of jobs comprises at least one job ~~configured~~ to read the file format and data structure of the known input file and convert that input file to a tab delimited file format or data structure that is common to the plurality of jobs, at least one job ~~configured~~ for extracting data from a file, at least one job ~~configured~~ for archiving files, at least one job ~~configured~~ for loading files, at least one job ~~configured~~ for messaging, at least one job ~~configured~~ for transforming data in a file, at least one job ~~configured~~ for validating data in a file, and further wherein each job includes an indicator of job termination success or job termination failure, and the plurality of jobs can be linked ~~are configured for linking~~ according to the indicator of job termination success or job termination failure.

13. (currently amended) A programmed computer for software application

development suitable for processing input files with varying file formats or data structures, comprising:

a memory having at least one region for storing computer executable program code; and

a processor for executing the program code stored in the memory, wherein the program code comprises:

code to characterize the file format and data structure of at least one known input file type; and

code to create a library comprising a plurality of jobs ~~configured~~ to perform predetermined functions, wherein the library of jobs comprises at least one job ~~configured~~ to read the file format and data structure of the known input file and convert that input file to a tab delimited file format or data structure that is common to the plurality of jobs, at least one job ~~configured~~ for extracting data from a file, at least one job ~~configured~~ for archiving files, at least one job ~~configured~~ for loading files, at least one job ~~configured~~ for messaging, at least one job ~~configured~~ for transforming data in a file, at least one job ~~configured~~ for validating data in a file, and further wherein each job includes an indicator of job termination success or job termination failure, and the plurality of jobs can be linked ~~are configured for linking~~ according to the indicator of job termination success or job termination failure.

14. (withdrawn) A method for processing a data file, the method comprising:  
selecting a plurality of jobs from a preexisting library of jobs, wherein the library of jobs comprises at least one job configured to read the file format and data structure of the known input file and convert that input file to a tab delimited file format or data structure that is

common to the plurality of jobs, at least one job configured for extracting data from a file, at least one job configured for archiving files, at least one job configured for loading files, at least one job configured for messaging, at least one job configured for transforming data in a file, at least one job configured for validating data in a file, and further wherein each job includes an indicator of job termination success or job termination failure;

creating a job stream, the job stream comprising the plurality of jobs linked to each other according to the indicator of job termination success or job termination failure;

creating a manager;

receiving the data file; and

processing the data file with the job stream in a plurality of steps, either in sequence or in parallel, using the manager, wherein at least one of the plurality of steps comprises converting the file format and data structure of the data file to the file format or data structure common to the plurality of jobs.

15. (withdrawn) A method according to claim 14, wherein receiving the data file comprises receiving the data file as an e-mail attachment.

16. (withdrawn) A method according to claim 14, wherein receiving the data file comprises receiving the data file using file transfer protocol.

17. (withdrawn) A method according to claim 14, wherein receiving the data file comprises receiving the data file using hypertext transfer protocol.

18. (withdrawn) A method according to claim 14, wherein receiving the data file further comprises checking security access for the data file.

19. (withdrawn) A method according to claim 14, further comprising creating a visual representation of the plurality of jobs and the interlinking of the jobs.

20. (withdrawn) A method according to claim 14, further comprising creating an extensible markup language document corresponding to the job stream.

21. (currently amended) A system implemented at least partially in a programmed computer for software application development, comprising a processor with:  
computer program means for selecting a plurality of jobs from a preexisting library of jobs, wherein the library of jobs comprises at least one job ~~configured~~ to read the file format and data structure of the known input file and convert that input file to a tab delimited file format or data structure that is common to the plurality of jobs, at least one job ~~configured~~ for extracting data from a file, at least one job ~~configured~~ for archiving files, at least one job ~~configured~~ for loading files, at least one job ~~configured~~ for messaging, at least one job ~~configured~~ for transforming data in a file, at least one job ~~configured~~ for validating data in a file, and further wherein each job includes an indicator of job termination success or job termination failure;

computer program means for creating a job stream, the job stream comprising the plurality of jobs linked to each other according to the indicator of job termination success or job termination failure;

computer program means for creating a manager;

computer program means for receiving the data file; and

computer program means for processing the data file with the job stream in a plurality of steps, either in sequence or in parallel, using the manager, wherein at least one of the plurality of steps comprises converting the file format and data structure of the data file to the tab delimited file format or data structure common to the plurality of jobs.

22. (cancelled)

23. (currently amended) A computer-executable program that is tangibly embodied on a computer-readable medium as having computer executable software code ~~stored thereon~~, the code for software application development, the code comprising:

code to select a plurality of jobs from a preexisting library of jobs, wherein the library of jobs comprises at least one job ~~configured~~ to read the file format and data structure of the known input file and convert that input file to a tab delimited file format or data structure that is common to the plurality of jobs, at least one job ~~configured~~ for extracting data from a file, at least one job ~~configured~~ for archiving files, at least one job ~~configured~~ for loading files, at least one job ~~configured~~ for messaging, at least one job ~~configured~~ for transforming data in a file, at least one job ~~configured~~ for validating data in a file, and further wherein each job includes an indicator of job termination success or job termination failure;

code to create a job stream, the job stream comprising the plurality of jobs linked to each other according to the indicator of job termination success or job termination failure;

code to create a manager;

code to receive the data file; and

code to process the data file with the job stream in a plurality of steps, either in sequence or in parallel, using the manager, wherein at least one of the plurality of steps comprises converting the file format and data structure of the data file to the tab delimited file format or data structure common to the plurality of jobs.

24. (currently amended) A programmed computer for software application development, comprising:

a memory having at least one region for storing computer executable program code; and

a processor for executing the program code stored in the memory, wherein the program code comprises:

code to select a plurality of jobs from a preexisting library of jobs, wherein the library of jobs comprises at least one job ~~configured~~ to read the file format and data structure of the known input file and convert that input file to a tab delimited file format or data structure that is common to the plurality of jobs, at least one job ~~configured~~ for extracting data from a file, at least one job ~~configured~~ for archiving files, at least one job ~~configured~~ for loading files, at least one job ~~configured~~ for messaging, at least one job ~~configured~~ for transforming data in a file, at least one job ~~configured~~ for validating data in a file, and further wherein each job includes an indicator of job termination success or job termination failure;

code to create a job stream, the job stream comprising the plurality of jobs linked to each other according to the indicator of job termination success or job termination failure;

code to create a manager;



code to receive the data file; and

code to process the data file with the job stream in a plurality of steps,  
either in sequence or in parallel, using the manager, wherein at least one of the plurality of steps  
comprises converting the file format and data structure of the data file to the tab delimited file  
format or data structure common to the plurality of jobs.

25 - 26. (cancelled)